

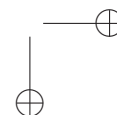
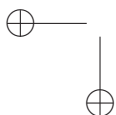
# طعم پایتون

## در برنامه نویسی

سید محمد جواد رضویان

علی رمزی، رضا علیمیرادی

دانشگاه قم



سرشناسه	: رضویان، سیدمحمدجواد، ۱۳۶۲
عنوان و نام پدیدآور	: طعم پایتون در برنامه‌نویسی/سیدمحمدجواد رضویان، علی رمزی، رضا علیمیرادی
مشخصات نشر	: قم: دانشگاه قم، انتشارات، ۱۳۹۷.
مشخصات ظاهری	: ۴۴۴ ص.: مصور، جدول، نمودار.
شابک	: ۹۷۸-۶۰۰-۸۴۳۶-۳۱-۷
وضعیت فهرست نویسی	: فیپا
یادداشت	: واژه نامه، کتاب‌نامه.
موضوع	: پایتون (زبان برنامه‌نویسی کامپیوتر) - مسائل، تمرین‌ها و غیره.
موضوع	: Python(Computer Programming Language)-Problems, Exercises, etc.
شناسه افزوده	: رمزی، علی، ۱۳۵۰ -
شناسه افزوده	: علیمیرادی، رضا، ۱۳۶۱ -
رده‌بندی کنگره	: ۱۳۹۷ ر۹۵۶/۷۳ پ QAV۶/۷۳
رده‌بندی دیویی	: ۰۰۵/۱۳۳
شماره کتابشناسی ملی	: ۵۳۶۹۷۳۵



انتشارات دانشگاه قم

عنوان کتاب: طعم پایتون در برنامه‌نویسی  
تألیف: سیدمحمدجواد رضویان، علی رمزی، رضا علیمیرادی  
چاپ و صحافی: هوشنگی  
ناظر فنی: علیرضا معظمی  
طراح جلد: ابوالفضل عبیدی  
صفحه آرا: سیدمحمدجواد رضویان  
تاریخ و نوبت چاپ: اول، پاییز ۱۳۹۷  
شمارگان: ۵۰۰

قیمت: ۳۵۰۰۰ تومان

ISBN: 978-600-8436-31-7

شابک: ۹۷۸-۶۰۰-۸۴۳۶-۳۱-۷

آدرس الکترونیکی: [publication@qom.ac.ir](mailto:publication@qom.ac.ir)

کلیه حقوق مادی و معنوی برای ناشر محفوظ است.

قم، بلوار الغدیر، دانشگاه قم، اداره چاپ و انتشارات دانشگاه  
تلفن: ۰۲۵-۳۲۱۰۳۳۴۴ - ۰۲۵-۳۲۱۰۳۳۴۵

## پیشگفتار

اولین درسی که دانشجویان رشته کامپیوتر با آن آشنا خواهند شد درس «مبانی کامپیوتر و برنامه‌سازی» است. به طور معمول این درس، ابتدا به مفاهیم پایه کامپیوتر، الگوریتم و حل مسئله می‌پردازد؛ سپس در ادامه درس دانشجویان با یادگیری یکی از زبان‌های برنامه‌نویسی سطح بالا، سعی می‌نمایند در قالب نحو آن زبان به حل مسائل بپردازند و به تدریج با اینکه چگونه به زبان کامپیوتر صحبت کنند آشنا شوند.

یکی از نکاتی که در این بین، بسیار مهم است، ساده و قابل فهم بودن زبانی است که دانشجویان باید با آن تعامل داشته باشند. طبیعی است هر چه یادگیری و فهم این زبان سخت‌تر باشد، دانشجویان به جای تمرکز بر روی حل مسئله و بیان الگوریتم به زبان کامپیوتر، باید درگیر جزئیات پیاده‌سازی در آن زبان شوند. البته این نظریه مخالفینی هم دارد که نگارندگان کتاب جز آن دسته نیستند.

نگارنده اول، سال ۱۳۸۹ پس از چندین سال تدریس برنامه‌نویسی به زبان‌های سی و سی‌پلاس‌پلاس بر همان مبنایی که توضیح داده شد تصمیم به تدریس زبان پایتون در دانشگاه قم می‌گیرد. خوشبختانه در تابستان همان سال با ابلاغ سرفصل‌های جدیدی از طرف وزارت علوم این زبان نیز جزء زبان‌های پیشنهادی برای آموزش به دانشجویان قرار می‌گیرد. اگر دانشگاه قم، دانشگاه پیشرو در تدریس زبان پایتون به عنوان اولین زبان برنامه‌نویسی به دانشجویان نباشد قطعاً جزء اولین دانشگاه‌هایی در ایران است که دست به چنین انتخابی زده‌اند هر چند که دانشگاه‌های برتر دنیا از جمله ام‌آی‌تی سال‌ها قبل، پایتون را برای آموزش به دانشجویان مبانی کامپیوتر برگزیده‌اند. مؤلف اول به سبب علاقه‌ای که به دنیای نرم‌افزارهای آزاد و به خصوص سیستم عامل لینوکس داشت از چند سال قبل‌تر با این زبان آشنا

شده و برنامه‌نویسی می‌نمود. به هر حال این تصمیم، تصمیم سختی بود و اگر حمایت‌های مدیریت محترم گروه علوم کامپیوتر جناب آقای دکتر شیردل در آن زمان نبود، چنین اتفاقی در دانشگاه قم رخ نمی‌داد. شاید ایشان و برخی فارغ‌التحصیلان آن دوره که اکنون این کتاب را ملاحظه می‌نمایند مقاومت سرسختانه دانشجویان را در یکی دو سال ابتدایی تدریس این درس به خاطر بیاورند.

از مشکلاتی که در آن دوره وجود داشت عدم دسترسی دانشجویان به منابع اصلی فارسی بود. از همان ابتدا از [۳] به عنوان منبع اصلی درس استفاده شد و در کنار آن در طول ترم به منابعی دیگر از جمله [۸، ۱۱، ۴] مراجعه می‌گردید. بی‌شک کتاب مزبور، یکی از خودآموزترین و فشرده‌ترین کتب آموزش پایتون در نوع خود است. از همان ابتدای امر، ترجمه این کتاب می‌توانست راهگشای بسیاری از دانشجویان گردد؛ گرچه هیچگاه نگارندگان فرصت این کار را نیافتند لکن امروزه از این کتاب، ترجمه‌ای فارسی در بازار موجود است و دیگر نیازی به اختراع مجدد چرخ نیست و البته امروز نسبت به آن سال، تعدادی زیادی کتاب فارسی در زمینه زبان پایتون و یا کاربردهای آن به فارسی ترجمه و یا تألیف شده است.

گذشت و گذشت تا حدود پنج سال پیش یکی از همکاران پیشنهاد تالیف کتابی برای این زبان را دادند و نگارندگان را نیز این پیشنهاد موافق آمد لیک آن روز، هیچگاه این تصور را نمی‌نمودند که چنین فرآیند طولانی مدتی را در پیش داشته باشند؛ بهر روی امروز این کتاب تمام شده است و پیش روی شماست.

هر چند که تصمیم ابتدایی در تدوین کتاب، اختصاص آن به دانشجویان رشته بایوانفورماتیک بود اما در ادامه، نگارندگان بر این نمط باقی نماندند؛ لیک همین تصمیم سبب می‌شود که منبع [۶] نقش پر رنگ‌تری نسبت به دیگر منابع داشته باشد و بسیاری مثال‌ها در فصول اولیه کتاب از حوزه این علم باشد ولی با این وجود مثال‌ها برای مابقی رشته‌ها نیز قابل فهم است؛ سپس در مرتبه بعد، منبع [۲] قرار دارد. ضمناً سعی کتاب بر این بوده است که با بیان مثال‌های بسیار، دانشجویان را در درک و فهم بهتر این زبان راهنمایی نماید. به همین سبب نیز بخش‌هایی در پیوست، علاوه بر متن اصلی خود کتاب به این هدف اختصاص داده شده است به خصوص آخرین پیوست کتاب که با بیان حجم زیادی مسأله و پاسخ‌های آن، این جنبه کتاب را تقویت می‌نماید؛ این مسائل غالباً برگرفته از [۱، ۵، ۹، ۱۰] است. جنبه‌های دیگری از پایتون که یادگیری آن‌ها برای علاقه‌مندان جذاب خواهد بود و نیز ابزارهایی کاربردی در پیوست‌ها گنجانده شده‌اند. روش این کتاب بر خلاف کتب معمول آموزش پایتون، بر پایه استفاده وسیع از مثال‌هایی کاربردی و معرفی جنبه‌هایی است که بیشتر مورد کاربرد است تا بیان تفصیلی تمامی جزئیات. کتاب حاضر همچنین می‌تواند به عنوان منبع درس

«مبانی کامپیوتر و برنامه‌سازی» قابل استفاده باشد اما با این حال برای دانشجویان ترم‌های بالاتر، و نیز علاقه‌مندان یادگیری پایتون، مطالبی مفید و کاربردی را شامل است.

این کتاب تماماً توسط نگارندگان با کمک سیستم حروفچینی لاتک و بسته‌ی زی‌پرشین که توسط جناب آقای دکتر وفا خلیقی تهیه شده آماده گردیده است؛ این سیستم برخلاف نرم‌افزار مایکروسافت ورد که در میان کاربران فارسی زبان کامپیوتر متداول است، رایگان و منبع باز بود و به راحتی حتی می‌توان امکاناتی را با کمک زبان برنامه‌نویسی تک/لاتک به آن افزود. گرچه باید اذعان نمود که امروزه این سیستم حروفچینی، در میان جامعه آکادمیک ایران نیز جایگاه خود را یافته است. بهر حال کیفیت حروفچینی این کتاب مدیون این سیستم است. طرح روی جلد نیز که با لاتک طراحی شده ممکن است برای دانشجویان رشته کامپیوتر کمی نامفهوم باشد اما تصویر، نشانگر جدول کدون‌های RNA به همراه آمینو اسیدهایی است که کدگذاری می‌نماید و بر گرفته از آدرس <http://www.texample.net/tikz/examples/rna-codons-table/> می‌باشد.

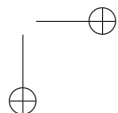
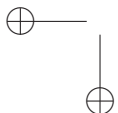
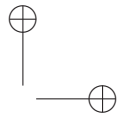
با وجود سعی و تلاشی که در جهت تکمیل این کتاب به عمل آمده است، بی‌شک عاری از اشتباه و نقصان نخواهد بود. لذا از کلیه پیشنهادهایی که در راستای رفع نواقص و تکمیل این کتاب ارائه شود استقبال خواهیم نمود. خواهشمند است پیشنهادات/انتقادات خود را به آدرس قم، دانشگاه قم، گروه علوم کامپیوتر و یا آدرس الکترونیکی [pythonbook.ir@gmail.com](mailto:pythonbook.ir@gmail.com) ارسال دارید.

سیدمحمدجواد رضویان،

علی رمزی و رضا علیمرادی

اعضای هیأت علمی دانشگاه قم

بهار ۱۳۹۷



# فهرست مطالب

۱	۱ مقدمه
۷	۲ انواع ساده و عبارات
۷	۱.۲ انواع ساده . . . . .
۸	۱.۱.۲ بولین . . . . .
۸	۲.۱.۲ عدد صحیح . . . . .
۹	۳.۱.۲ عدد حقیقی . . . . .
۱۰	۴.۱.۲ رشته . . . . .
۱۱	۲.۲ عبارت . . . . .
۱۱	۱.۲.۲ عملگرهای عددی . . . . .
۱۳	۲.۲.۲ عملگرهای منطقی . . . . .
۱۴	۳.۲.۲ عملگرهای رشته . . . . .
۱۸	۴.۲.۲ فراخوانی‌ها . . . . .
۲۲	۵.۲.۲ عبارات مرکب . . . . .
۲۵	۳ نام‌ها، توابع و ماژول‌ها



۱.۳	انتساب نام‌ها	۲۶
۲.۳	تعریف توابع	۲۷
۱.۲.۳	پارامترهای تابع	۳۰
۲.۲.۳	توضیحات و مستندات	۳۱
۳.۲.۳	تأکیدات	۳۲
۴.۲.۳	مقادیر پیش‌فرض پارامترها	۳۴
۳.۳	استفاده از ماژول‌ها	۳۶
۱.۳.۳	وارد کردن	۳۶
۲.۳.۳	فایل‌های پایتون	۴۰

#### ۴ گردایه‌ها

۱.۴	مجموعه‌ها	۴۳
۲.۴	دنباله‌ها	۴۴
۱.۲.۴	رشته‌ها، بایت‌ها، آرایه‌های بایتی	۴۷
۲.۲.۴	محدوده‌ها	۴۹
۳.۲.۴	تاپل‌ها	۵۶
۴.۲.۴	لیست‌ها	۵۸
۳.۴	نگاشت‌ها	۵۹
۱.۳.۴	فرهنگ داده‌ها	۶۴
۴.۴	جریان‌ها	۶۴
۱.۴.۴	فایل‌ها	۶۹
۲.۴.۴	مولدها	۷۰
۵.۴	عبارات مرتبط با گردایه	۷۵
۱.۵.۴	ادراکات	۷۶
۲.۵.۴	پارامترهای تابعی	۷۷

#### ۵ دستورات کنترلی

۱.۵	شرط‌ها	۹۳
۲.۵	حلقه‌ها	۹۵

۹۹	مثال‌هایی از حلقه‌های ساده	۱.۲.۵
۹۹	مقداردهی اولیه مقادیر حلقه	۲.۲.۵
۱۰۱	حلقه بی‌پایان	۳.۲.۵
۱۰۲	حلقه با شرایط نگهبان	۴.۲.۵
۱۰۴	تکرارها	۳.۵
۱۰۵	دستورات تکرار	۱.۳.۵
۱۰۷	انواع تکرار	۲.۳.۵
۱۲۷	کنترل‌کننده‌های خطا	۴.۵
۱۲۹	خطاهای پایتون	۱.۴.۵
۱۳۱	دستورات کنترل‌کننده خطا	۲.۴.۵
۱۳۵	بروز استثناءها	۳.۴.۵
۱۳۷	مثال‌های گسترده‌تر	۵.۵
۱۳۷	پارسر فایل Grand Unified Bioinformatics	۱.۵.۵
۱۳۹	تجزیه‌نمودن فایل‌های GenBank	۲.۵.۵
۱۴۲	ترجمه توالی‌های RNA	۳.۵.۵
۱۴۵	ساخت جدول از یک فایل متنی	۴.۵.۵
۱۵۱	۶ کلاس‌ها	
۱۵۲	تعریف کلاس‌ها	۱.۶
۱۵۴	خصوصیات نمونه	۱.۱.۶
۱۶۶	خصوصیات کلاس	۲.۱.۶
۱۷۳	روابط کلاس و متد	۲.۶
۱۷۴	تجزیه	۱.۲.۶
۱۸۲	ارث‌بری	۲.۲.۶
۱۹۵	۷ ابزارهای کاربردی	
۱۹۵	محیط سیستم	۱.۷
۱۹۶	تاریخ و زمان: datetime	۱.۱.۷
۱۹۹	اطلاعات سیستم	۲.۱.۷

۲۰۴	ابزارهای کاربردی خط فرمان	۳.۱.۷
۲۱۱	ارتباطات	۴.۱.۷
۲۱۴	فایل سیستم	۲.۷
۲۱۴	واسط سیستم عامل: os	۱.۲.۷
۲۱۸	دستکاری مسیرها: os.path	۲.۲.۷
۲۲۱	توسعه نام فایل: glob و fnmatch	۳.۲.۷
۲۲۳	برنامه‌های کاربردی پوسته: shutil	۴.۲.۷
۲۲۴	مقایسه فایل‌ها و دایرکتوری‌ها	۵.۲.۷
۲۲۸	کار با متون	۳.۷
۲۲۸	قالب‌بندی بلاک‌های متن: textwrap	۱.۳.۷
۲۳۰	ابزارهای کاربردی رشته: string	۲.۳.۷
۲۳۱	فرمت‌های جداسازده با تب و کاما: csv	۳.۳.۷
۲۳۳	خواندن و نوشتن بر پایه رشته: io	۴.۳.۷
۲۳۴	رسانه ذخیره‌سازی دائمی	۴.۷
۲۳۴	متون دائمی: dbm	۱.۴.۷
۲۳۷	اشیاء دائمی: pickle	۲.۴.۷
۲۳۹	شیء ذخیره‌سازی دائمی کلیددار: shelve	۳.۴.۷
۲۴۰	ابزارهای اشکال‌زدایی	۴.۴.۷

## ۸ تطابق الگو

۲۴۵	نحو پایه	۱.۸
۲۴۶	تطابق با طول ثابت	۱.۱.۸
۲۴۷	تطابق با طول متغیر	۲.۱.۸
۲۵۱	تطابق حریصانه در مقایسه با تطابق غیرحریصانه	۳.۱.۸
۲۵۲	گروه‌بندی و تفکیک	۴.۱.۸
۲۵۳	اعمال ماژول re	۲.۸
۲۵۴	توابع	۱.۲.۸
۲۵۴	فلگ‌ها	۲.۲.۸

۲۵۷	متدها	۳.۲.۸
۲۵۸	نتایج توابع و متدهای re	۳.۸
۲۵۸	فیلدهای شیء تطابق	۱.۳.۸
۲۵۹	متدهای شیء تطابق	۲.۳.۸
۲۵۹	کاربردهایی از ماژول re	۴.۸
۲۶۰	چند مثال کوچک و سریع	۱.۴.۸
۲۶۲	استخراج توصیفات از فایل های توالی	۲.۴.۸
۲۶۴	استخراج واردها از فایل های توالی	۳.۴.۸
۲۷۳	مثال های بیشتر	۹
۲۷۳	رشته ها و متن ها	۱.۹
۲۷۳	اتصال رشته	۱.۱.۹
۲۷۴	مقایسه رشته ها	۲.۱.۹
۲۷۵	جستجوی زیررشته	۳.۱.۹
۲۷۵	استخراج زیررشته	۴.۱.۹
۲۷۶	جایگزینی بخش تطابق یافته رشته	۵.۱.۹
۲۷۶	معکوس نمودن رشته	۶.۱.۹
۲۷۶	پیراستن فضاهاى سفید	۷.۱.۹
۲۷۷	تغییر کوچکی و بزرگی حروف	۸.۱.۹
۲۷۷	تبدیل به عدد	۹.۱.۹
۲۷۷	تکرار روی حروف تشکیل دهنده یک رشته	۱۰.۱.۹
۲۷۸	آماره های روی رشته	۱۱.۱.۹
۲۷۸	ترجمه	۱۲.۱.۹
۲۷۸	عدد، تاریخ و زمان	۲.۹
۲۷۹	ساخت عدد صحیح	۱.۲.۹
۲۷۹	ایجاد اعداد حقیقی	۲.۲.۹
۲۷۹	گردنمودن اعداد حقیقی به اعداد صحیح	۳.۲.۹
۲۸۰	قالب بندی اعداد	۴.۲.۹

۲۸۱	کار با اعدادی با دقت دلخواه	۵.۲.۹
۲۸۲	تولید اعداد تصادفی	۶.۲.۹
۲۸۲	بدست آوردن تاریخ و زمان جاری سیستم	۷.۲.۹
۲۸۳	محاسبه تفاوت تاریخ/زمان	۸.۲.۹
۲۸۳	قالب‌بندی تاریخ و زمان	۹.۲.۹
۲۸۳	خواندن تاریخ و زمان از یک رشته	۱۰.۲.۹
۲۸۴	تکرارپذیرها و مولدها	۳.۹
۲۸۵	تکرار روی محتوای یک لیست	۱.۳.۹
۲۸۵	استخراج محتویات یک تکرارپذیر	۲.۳.۹
۲۸۶	فیلترنمودن یک تکرارپذیر	۳.۳.۹
۲۸۶	تکرار روی محتویات یک فایل	۴.۳.۹
۲۸۷	تکرار روی داده‌هایی که تکرارپذیر ندارند	۵.۳.۹
۲۸۸	ساخت کلاس‌های استاندارد تکرارپذیرها	۶.۳.۹
۲۸۸	فایل‌های ورودی/خروجی	۴.۹
۲۸۹	کپی کردن فایل	۱.۴.۹
۲۸۹	جابجا نمودن فایل	۲.۴.۹
۲۹۰	خواندن و نوشتن فایل‌های متنی	۳.۴.۹
۲۹۱	خواندن و نوشتن فایل‌های XML	۴.۴.۹
۲۹۲	ایجاد دایرکتوری	۵.۴.۹
۲۹۲	چک نمودن یک دایرکتوری برای تغییرات	۶.۴.۹
۲۹۳	تکرار روی فایل‌های یک دایرکتوری	۷.۴.۹
۲۹۳	ذخیره نموده اشیاء داده‌ای	۸.۴.۹
۲۹۴	فشرده‌سازی فایل‌ها	۹.۴.۹
۲۹۵	توابعی	۵.۹
۲۹۵	ایجاد توابع پایه	۱.۵.۹
۲۹۵	استفاده از پارامترهای نامدار بجای پارامترهای مکانی	۲.۵.۹
۲۹۶	استفاده از مقادیر پیش‌فرض در توابع	۳.۵.۹
۲۹۷	پیاده‌سازی الگوریتم‌های بازگشتی	۴.۵.۹

۵.۵.۹	استفاده از حساب لاندای برای توابع موقتی بی‌نام	۲۹۷
۶.۵.۹	تولید توابع بخصوص	۲۹۸
۶.۹	کلاس‌ها و اشیاء	۲۹۸
۱.۶.۹	تشخیص نوع یک شیء (در پایتون هر چیزی یک شیء است)	۲۹۸
۲.۶.۹	ساخت کلاس	۲۹۹
۳.۶.۹	افزودن فیلدهای خصوصی	۲۹۹
۴.۶.۹	زیرکلاس	۳۰۰
۵.۶.۹	مقداردهی اولیه اشیاء	۳۰۱
۶.۶.۹	مقایسه اشیاء	۳۰۲
۷.۶.۹	تغییر یک شیء پس از ایجاد	۳۰۲
۸.۶.۹	پیاده‌سازی رفتار چندریخت	۳۰۳
۷.۹	محاسبات عددی و ماژول numpy	۳۰۳
۱.۷.۹	کپی یک آرایه	۳۰۵
۲.۷.۹	دسترسی به داده‌های آرایه	۳۰۶
۳.۷.۹	دستکاری یک ماتریس	۳۰۷
۴.۷.۹	محاسبه تبدیل سریع فوریه	۳۰۷
۵.۷.۹	بارگذاری داده در یک آرایه	۳۰۸
۶.۷.۹	ذخیره نمودن آرایه‌ها	۳۰۸
۷.۷.۹	تولید اعداد تصادفی	۳۰۹
۸.۷.۹	محاسبات پایه آماری	۳۰۹
۹.۷.۹	محاسبه هیستوگرام	۳۱۰

## پیوست‌ها

### الف کلمات کلیدی پایتون

### ب جدول اسکی

۱.ب	تاریخچه اسکی	۳۱۳
۲.ب	اسکی چیست؟	۳۱۴

ب.۳ هنراسکی ..... ۳۱۵

## پ تحلیل داده‌ها با pandas ۳۱۷

پ.۱ کار با داده‌های تک بعدی ..... ۳۱۷

پ.۲ کار با داده‌های دو بعدی ..... ۳۱۸

پ.۳ کار با داده‌های سه بعدی ..... ۳۱۹

پ.۴ واردنمودن داده‌ها از فایل‌های CSV ..... ۳۲۰

پ.۵ ذخیره در یک فایل CSV ..... ۳۲۰

پ.۶ واردنمودن داده‌ها از صفحه‌گسترده ..... ۳۲۰

پ.۷ ذخیره داده‌ها در یک فایل صفحه‌گسترده ..... ۳۲۱

پ.۸ گرفتن داده‌هایی ابتدایی و انتهایی ..... ۳۲۱

پ.۹ تلخیص داده‌ها ..... ۳۲۲

پ.۱۰ مرتب‌سازی داده‌ها ..... ۳۲۲

پ.۱۱ اعمال توابع روی سطرها یا ستون‌ها ..... ۳۲۳

پ.۱۲ اعمال توابع روی تک‌تک عناصر ..... ۳۲۴

پ.۱۳ تکرار روی داده‌ها ..... ۳۲۴

## ت فرابرنامه‌نویسی ۳۲۷

ت.۱ استفاده از آذینگر تابع به منظور پوشاندن کد موجود ..... ۳۲۷

ت.۲ نوشتن آذینگر تابع به منظور پوشاندن کد موجود ..... ۳۲۸

ت.۱.۲ بازنمودن یک تابع آذین‌شده ..... ۳۲۸

ت.۳ استفاده از فوق‌کلاس‌ها به منظور تغییر ساختار یک کلاس ..... ۳۲۹

ت.۴ نوشتن یک فوق‌کلاس ..... ۳۲۹

ت.۵ استفاده از امضاء به منظور تغییر پارامترهایی که یک تابع می‌پذیرد ..... ۳۳۰

## ث همروندی ۳۳۳

ث.۱ ایجاد یک نخ ..... ۳۳۳

ث.۲ استفاده از قفل‌ها ..... ۳۳۴

ث.۳ تنظیم یک مانع ..... ۳۳۵

۴.ث	ایجاد یک فرآیند . . . . .	۳۳۵
۵.ث	ارتباط بین فرآیندها . . . . .	۳۳۵
۶.ث	ایجاد مخزنی از عملگرها . . . . .	۳۳۶
۷.ث	ایجاد زیرفرآیند . . . . .	۳۳۷
۸.ث	زمان‌بندی وقایع . . . . .	۳۳۷

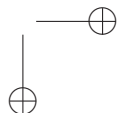
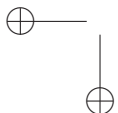
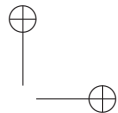
ج	بسته‌ها و ماژول‌ها	۳۳۹
ج.۱	واردنمودن ماژول‌ها . . . . .	۳۳۹
ج.۲	نصب ماژول‌ها از روی کد منبع . . . . .	۳۴۰
ج.۳	نصب ماژول‌ها از روی Pypi . . . . .	۳۴۰
ج.۴	بروزرسانی یک ماژول با استفاده pip . . . . .	۳۴۱

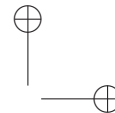
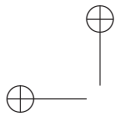
چ	ابزارهای کاربردی	۳۴۳
چ.۱	ایجاد یک محیط مجازی . . . . .	۳۴۳
چ.۲	استفاده از پوسته IPython . . . . .	۳۴۴
چ.۳	استفاده از محیط Jupyter . . . . .	۳۴۹
چ.۴	xonsh جایگزینی برای پوسته خط فرمان . . . . .	۳۵۱

ح	تست برنامه و اشکال‌زدایی	۳۵۳
ح.۱	زمان‌گیری بخشی از کد . . . . .	۳۵۳
ح.۲	نمایش‌سازی کد . . . . .	۳۵۴
ح.۳	ردگیری زیرروال‌ها . . . . .	۳۵۵
ح.۴	ردگیری تخصیصات حافظه . . . . .	۳۵۶
ح.۵	اجرای تست‌های یگه . . . . .	۳۵۷
ح.۶	اشکال‌زدایی کد . . . . .	۳۵۷

خ	تمارین	۳۵۹
---	--------	-----







## مقدمه

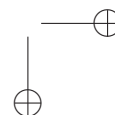
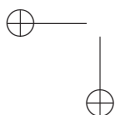
همان‌طوری که می‌دانید زبان‌های برنامه‌نویسی زیادی با مقاصد گوناگونی طراحی شده‌اند اما در غالب موارد، برنامه‌نویسان حرفه‌ای در تمامی مدت فعالیتشان، تنها از یک زبان برنامه‌نویسی و یا شاید چند زبان محدود بهره‌برند لذا در بسیاری موارد نسبت به دیگر زبان‌های برنامه‌نویسی و تفاوت‌های میان آن‌ها آگاهی مناسبی ندارند. به‌طور معمول زبان‌های برنامه‌نویسی در یکی از چهار الگوی زیر قرار می‌گیرند هر چند که ممکن است زبانی، از چند الگوی ذیل پشتیبانی نماید لکن در همان زبان نیز یک الگو، غالب خواهد بود. و اما الگوها به شرح ذیل هستند: (۱) رویه‌ای<sup>۱</sup> یا امری<sup>۲</sup> (۲) توصیفی<sup>۳</sup> (۳) تابعی<sup>۴</sup> شیء‌گرایی<sup>۵</sup>.

**رویه‌ای یا امری** در این شیوه که رایج‌ترین مدل زبان‌های برنامه‌نویسی است، محاسبات در قالب یک سری از دستورات بیان شده و توسط کامپیوتر اجرا می‌گردد. این الگو، مکانیزم‌هایی برای شرط، تکرار، فراخوانی زیرروال‌ها و ... فراهم می‌آورد. تاریخ این مدل به اولین روزهای کامپیوتر باز می‌گردد و هسته اصلی اکثر زبان‌های برنامه‌نویسی مدرن است حتی آن‌هایی که برای الگوهای دیگر برنامه‌نویسی طراحی شده‌اند.

**توصیفی** برنامه‌نویسی توصیفی بر پایه بیان حقایق و استنتاج‌های منطقی که حقایق دیگر از آن‌ها منتج می‌گردد قرار دارد. تجسم اصلی این مدل برنامه‌نویسی، زبان پرولوگ<sup>۶</sup> است که به‌طور

---

<sup>۱</sup>procedural <sup>۲</sup>imperative <sup>۳</sup>declarative <sup>۴</sup>functional <sup>۵</sup>object-oriented <sup>۶</sup>Prolog



گسترده‌ای در تحقیقات هوش مصنوعی و کاربردهای آن استفاده می‌شود و از اوایل دهه ۸۰ میلادی مطرح گردیده است. پرولوگ به عنوان زبانی که از اساس بر پایه منطق است، محاسبات را به صورت یک سری محمولات بیان می‌دارد به گونه‌ای که این محمولات پازلی را برای سیستم به وجود می‌آورند تا آن را حل نماید.

**تابعی** در مدل برنامه‌نویسی تابعی، تمام محاسبات به صورت فراخوانی‌های توابع بیان می‌گردد. در آن زبان برنامه‌نویسی که مدل تابعی محض باشد، حتی دستور انتساب نیز وجود ندارد و مقادیر را تنها می‌توان از طریق پارامترهای توابع بکار برد. لیسپ<sup>۷</sup> اولین زبان برنامه‌نویسی تابعی است که به سال ۱۹۵۸ باز می‌گردد و نام آن از «زبان پردازش لیست»<sup>۸</sup> گرفته شده که اشاره‌ای به نوع ساختمان داده‌ای دارد که زبان بر پایه آن است. در دهه ۶۰ میلادی لیسپ زبان متدوال در هوش مصنوعی بود و هنوز نیز در تحقیقات هوش مصنوعی و کاربردهای آن نیز نقش پررنگی دارد. گرچه لیسپ دو دهه بعد، تغییرات بنیادین زیادی از جمله پشتیبانی از مدل شیء‌گرایی داشت. از جمله دیگر زبان‌های برنامه‌نویسی تابعی می‌توان به ام‌ال<sup>۹</sup> و هسکل<sup>۱۰</sup> اشاره داشت.

**شیء‌گرایی** الگوی برنامه‌نویسی شیء‌گرا در اواخر دهه ۶۰ میلادی ایجاد گردید و سپس طی دهه بعدی در محیط‌های تحقیقاتی توسعه داده شد و نهایتاً در دهه ۸۰ میلادی به طور گسترده‌ای در محیط‌های تجاری و آکادمیک بکار گرفته شد که از جمله این زبان‌ها می‌توان به اسمالتاک<sup>۱۱</sup>، آبجکتیوسی<sup>۱۲</sup> و سی‌پلاس‌پلاس<sup>۱۳</sup> اشاره داشت. این مدل در دهه بعد، بخش کلیدی در توسعه نرم‌افزارهای مدرن گردید. در همین دهه زبان جاوا<sup>۱۴</sup> نیز معرفی گردید و سیستم عامل مک<sup>۱۵</sup> روی یک هسته یونیکسی<sup>۱۶</sup> ساخته شد که در سطوح بالایی خود، بخصوص بخش واسط کاربری‌اش از آبجکتیوسی استفاده می‌نمود. جاوا اسکریپت<sup>۱۷</sup> که در صفحات وب به صورت گسترده استفاده می‌شود نیز از همین مدل استفاده می‌نماید. امروزه الگوی غالب، همین مدل برنامه‌نویسی شیء‌گرا است.

جنبه دیگری که زبان‌های برنامه‌سازی را از یکدیگر تمیز می‌دهد کاربرد اصلی این زبان‌ها است. برای مثال زبانی هدفش ساده بودن یادگیری آن و زبانی دیگر به منظور استفاده در سیستم‌های توکار طراحی شده است و یا دلایل مختلف دیگر. مجموعه این عوامل سبب می‌گردد که اقبال یا ادباری نسبت به یک زبان رخ دهد.

<sup>7</sup>LISP <sup>8</sup>LISt Processing language <sup>9</sup>ML <sup>10</sup>Haskell <sup>11</sup>Smalltalk <sup>12</sup>Objective-C <sup>13</sup>C++ <sup>14</sup>Java

<sup>15</sup>Mac OS X <sup>16</sup>Unix-like kernel <sup>17</sup>JavaScript

از تمامی این بحث‌ها که درگذریم زبان برنامه‌نویسی پایتون، زبانی بسیار زیبا است و برای همه چیز از آموزش به برنامه‌نویسان جدید تا مطالعات پیشرفته علمی، کارا و قابل استفاده است. به خصوص در سال‌های اخیر پایتون جایگاه ویژه‌ای در میان دانشمندان علوم مختلف یافته است و این محبوبیت نیز روز به روز رو به رشد است و کتابخانه‌های بسیاری در کاربردهای گوناگونی برای آن نوشته شده است. شاید سبب این توجه روزافزون به سوی پایتون سادگی و در عین حال قدرتمندی آن باشد.

پایتون در اواخر دهه ۸۰ میلادی توسط برنامه‌نویس هلندی، خیدو ون راسم<sup>۱۸</sup> ایجاد می‌گردد. به طور خلاصه می‌توان آن را زبانی تفسیری<sup>۱۹</sup> همه منظوره و سطح بالا دانست که به طور گسترده‌ای استفاده می‌گردد. فلسفه طراحی آن تاکید زیادی بر خوانایی کد دارد و نحو آن نیز به گونه‌ای است که برنامه‌نویس می‌تواند منظورش را با تعداد خطوط کمتری نسبت به زبان‌هایی مانند سی بیان دارد. در واقع هدف اولیه آقای راسم طراحی زبانی بود که قدرت سی را همراه با سادگی و راحتی شل اسکریپت<sup>۲۰</sup> در خود داشته باشد. پایتون از زمره زبان‌های با نوع پویا<sup>۲۱</sup> است بدین معنی که تا زمان انقیاد<sup>۲۲</sup>، نوع داده تعیین نمی‌گردد و در ضمن در طول برنامه نیز می‌تواند بارها و بارها تغییر کند؛ در اصطلاح زبان‌های برنامه‌سازی، پایتون جزء زبان‌های با تقیّد دیر هنگام<sup>۲۳</sup> است.

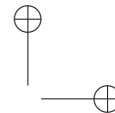
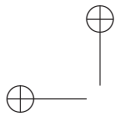
مفسر پایتون در بسیاری از سکوها<sup>۲۴</sup> رایج توسعه داده شده است و همین سبب می‌گردد که کد پایتون بدون هیچ نگرانی در سیستم‌های عامل متنوعی قابل حمل<sup>۲۵</sup> باشد.

از جمله ویژگی‌های نحوی زبان پایتون این است که هر دستور با انتهای خط تمام می‌شود و بدنه دستورات مرکب با یک حاشیه نسبت به خط سرآمد آن دستور مشخص می‌گردند؛ اگر با دیگر زبان‌های برنامه‌نویسی آشنایی داشته باشید می‌دانید که در بسیاری از این زبان‌ها از سمیکولن یا یک کلمه کلیدی و مشابهات آن برای نشان دادن پایان دستور یا بلاک دستورات استفاده می‌گردد؛ البته که برنامه‌نویسان هیچ نیازی به این علائم نداشته و در واقع این علائم را برای کمک به ترجمه در نحو آن زبان‌ها قرار داده‌اند. خود این نحو پایتون مقدار زیادی به خوانایی کد می‌افزاید و نیز جنبه‌های دیگری در نحو این زبان سبب می‌گردد تا برنامه‌نویس بیشتر تمرکزش معطوف به نتیجه تا جزئیات پیاده‌سازی باشد.

ساختار اصلی پایتون همان مدل امری یا رویه‌ای است که به طور معناداری تحت تاثیر برنامه‌نویسی تابعی است اما در اساس پایتون یک زبان برنامه‌نویسی شیء‌گرا است—لکن هیچ جزء برنامه‌نویسی توصیفی در آن وجود ندارد. تعداد کمی زبان برنامه‌نویسی است که چنین ترکیب زیبا و یکپارچه‌ای

<sup>18</sup>Guido van Rossum <sup>19</sup>interpreted language <sup>20</sup>shell script <sup>21</sup>dynamic type <sup>22</sup>binding <sup>23</sup>late binding

<sup>24</sup>platform <sup>25</sup>portable

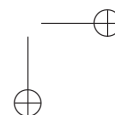
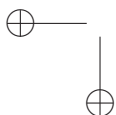


از الگوهای مختلف برنامه‌نویسی را در خود داشته باشد. در اواخر سال ۲۰۰۰ پایتون ۲/۰ عرضه گردید که ویژگی‌های مهم جدیدی از جمله پشتیبانی از یونیکد و یک زباله روب<sup>۲۶</sup> را در خود داشت. اواخر سال ۲۰۰۸ نیز پایتون ۳/۰ با تغییراتی گسترده و بدون پشتیبانی<sup>۲۷</sup> نسخه‌های قبلی ارائه گردید. اگر چه، پس از مدتی طولانی تست و بررسی نهایتاً بسیاری از ویژگی‌های آن با پایتون ۲/۶ و ۲/۷ سازگار گردید. پایان پشتیبانی از پایتون ۲.x در ابتدا تا سال ۲۰۱۵ تعیین گردید هر چند بعدها به ۲۰۲۰ موکول شد و سبب، حجم بسیار زیادی از کدهایی بود که با این نسخه از پایتون نگاشته شده بود و به آسانی امکان حمل آن‌ها به نسخه ۳ وجود نداشت. طراحی پایتون به گونه‌ای است که پشتیبانی خیلی خوبی از مدل برنامه‌نویسی تابعی را فراهم می‌آورد: توابع `map()`، `reduce()` و `filter()`؛ ادراکات لیست<sup>۲۸</sup>، فرهنگ‌داده‌ها<sup>۲۹</sup> و مجموعه‌ها<sup>۳۰</sup>؛ و نیز عبارات مولد<sup>۳۱</sup> را در خود دارد. همچنین دو کتابخانه استاندارد `itertools` و `functools` ابزارهای تابعی مشابه هسکل و ام‌ال استاندارد را پیاده‌سازی می‌نماید. فلسفه اصلی پایتون در PEP 20<sup>۳۲</sup> به صورت چند شعار آمده است که برخی از آن‌ها را در زیر نقل می‌نماییم.

- زیبایی بهتر از زشتی است.
- تصریح بهتر از بیان ضمنی است.
- سادگی بهتر از پیچیدگی است.
- پیچیدگی بهتر از مغلق بودن است.
- خوانایی مهم‌تر است.

<sup>۲۶</sup>garbage collector: زباله روبی یکی از مدل‌های خودکار مدیریت حافظه است. زباله روب سعی می‌نماید تا آن بخش‌هایی از حافظه را که توسط اشیائی که دیگر در برنامه استفاده نمی‌شوند آزاد گرداند. این ایده اولین بار توسط جان ماک‌کارتی (John McCarthy) به منظور ساده‌سازی مدیریت دستی حافظه در لیسپ در حدود سال ۱۹۵۹ طراحی گردید. <sup>۳۲</sup>پیشنهادهای این شماره—از مجموعه پیشنهادهاى بهبود دهنده پایتون (Python Enhancement Proposals)— که به تفکر پایتون (Zen of Python) نیز مشهور است شامل ۲۰ اصل کلی برنامه‌نویسی است که طراحی پایتون را تحت تأثیر خود قرار داده است. این پیشنهادها از طریق ویسایت رسمی پایتون در دسترس است. همچنین می‌توانید با تایپ دستور `import this` در مفسر پایتون این جملات را مشاهده نمایید.

<sup>27</sup>backwards-incompatible <sup>28</sup>list comprehensions <sup>29</sup>dictionaries <sup>30</sup>sets <sup>31</sup>generator expression



هسته اصلی پایتون خیلی کوچک است و در عین حال طراحی آن، قابلیت توسعه دارد و کتابخانه‌های بسیاری برای آن نوشته شده است. و همین هسته کوچک سبب می‌گردد که براحتی بتوان آن را به سکوها‌ی جدید نیز حمل نمود.

از سال ۲۰۰۳ پایتون جزء ۱۰ زبان اول برنامه‌نویسی قرار گرفته است. مارس ۲۰۱۷ پایتون به پله پنجم صعود نمود. از سال ۲۰۰۷ تا ۲۰۱۰ به عنوان زبان سال برنامه‌نویسی انتخاب گردید. تجربه نشان داده است در مسائلی که با رشته‌ها سر و کار دارد و یا در یک فرهنگ‌داده به جستجو می‌پردازد استفاده از زبان‌های اسکریپتی زودتر منتهی به جواب می‌گردد. هر چند که مصرف حافظه‌ای پایتون بهتر از جاوا است لکن در درجه‌ای بعد از سی و سی‌پلاس‌پلاس قرار دارد.

سازمان‌های بزرگی نظیر ویکی‌پدیا<sup>۳۳</sup>، گوگل<sup>۳۴</sup>، یاهو<sup>۳۵</sup>، سرن<sup>۳۶</sup>، ناسا<sup>۳۷</sup> و بسیاری دیگر نیز این زبان را بکار می‌بندند. ضمناً سایت‌های بسیاری از جمله شبکه اجتماعی خبری ردیت<sup>۳۸</sup> نیز تماماً با پایتون نوشته شده است. فریم‌ورک‌های متنوعی نیز برای این زبان آماده شده است که از جمله آن می‌توان به جنگو<sup>۳۹</sup> برای وب و یا توئیستد<sup>۴۰</sup> به منظور ارتباط بین کامپیوتری اشاره داشت که مورد اخیر توسط شرکت دراپ باکس<sup>۴۱</sup> استفاده می‌گردد.

کتابخانه‌هایی نظیر NumPy، SciPy، Matplotlib امکان کاربرد کارای پایتون را در محاسبات علمی فراهم می‌آورد. همچنین کتابخانه‌هایی تخصصی نظیر Biopython و Astropy کاربردهایی را در حیطه‌های مشخصی ممکن می‌سازند.

در بسیاری نرم‌افزارها پایتون به عنوان زبان اسکریپت‌نویسی استفاده شده است که از آن جمله می‌توان به 3D Max، Blender و Inkscape اشاره داشت. پایتون جزء تفکیک ناپذیر بسیاری از سیستم‌عامل‌های لینوکسی است به نحوی که حتی برخی از آن‌ها برنامه مدیریت بسته‌هایشان را به این زبان پیاده‌سازی نموده‌اند. حتی در برد رزبری پای<sup>۴۲</sup> که یک کامپیوتر با اندازه یک کارت ویزیت است، پایتون به عنوان زبان اصلی برنامه‌نویسی آن تطبیق داده شده است. پس از پایتون برخی زبان‌های برنامه‌نویسی تحت تاثیر این زبان بوده‌اند از آن جمله می‌توان به گو<sup>۴۳</sup>، گرووی<sup>۴۴</sup>، روبی<sup>۴۵</sup> و سوئیف<sup>۴۶</sup> اشاره داشت.

برای نصب پایتون کافی است که به سایت رسمی آن [۱۲] مراجعه نموده و نسخه‌ای منطبق با سیستم‌عامل خودتان را دانلود و سپس نصب نمایید.

<sup>33</sup>Wikipedia <sup>34</sup>Google <sup>35</sup>Yahoo <sup>36</sup>CERN <sup>37</sup>NASA <sup>38</sup>Reddit <sup>39</sup>Django <sup>40</sup>Twisted <sup>41</sup>Dropbox

<sup>42</sup>Raspberry Pi <sup>43</sup>Go <sup>44</sup>Groovy <sup>45</sup>Ruby <sup>46</sup>swift